Kadane's Algorithm - Step Notes

This note explains the logic of Kadane's Algorithm to find the maximum subarray sum. The algorithm works by iterating over the array, keeping track of the current sum (currSum) and updating the maximum sum found so far (maxSum).

Step-by-step rules:

- +ve + +ve => gives +ve: Add and continue, as the sum is increasing.
- -ve (small negative) + +ve => gives +ve: Add and continue, as the positive number can overcome the small loss.
- +ve + -ve (big negative) => gives -ve: Reset currSum to 0, because continuing would decrease the total.

Whenever currSum becomes negative, it means the subarray so far will only decrease the total sum of any future subarray. In such a case, we reset currSum to 0 and start a new subarray from the next element.

Pseudocode:

```
currSum = 0
maxSum = nums[0]

for num in nums:
    currSum += num
    maxSum = max(maxSum, currSum)
    if currSum < 0:
        currSum = 0
return maxSum
```